Privacy-aware transmission scheme based on homomorphic proxy re-encryption for NDN

Xian Guo*, Cheng Chen, Ma-Jiang Zhang, Aristide Ngaboyindekwe and Lai-Cheng Cao

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
Email: iamxg@163.com
Email: ccabner@163.com
Email: zmajiang@163.com
Email: bonheur235@yahoo.fr
Email: caolch@lut.cn
*Corresponding author

Abstract: Named data networking (NDN) is a data-centric Internet architecture. Although some security mechanisms are introduced in NDN packets, security still is a significant problem. We propose a privacy-aware transmission scheme for NDN (PATS_NDN) based on homomorphism and proxy re-encryption. To resolve name privacy in PATS_NDN, a content consumer can subscribe an interested content by using a blinded alias of content, and a data source can publish the produced content by using a blinded name. The alias of a content can be got from a quadratic function. A blinding algorithm is used on a content name and alias. In addition, homomorphic and proxy re-encryption are used to achieve secure content transmission. It has been proved that only a legitimate user can publish and access content on a network in our novel scheme. Finally, we analyse security attributes of our scheme and make a simple comparison with other related schemes.

Keywords: content-centric networking; named data networking; homomorphic encryption; PRE; proxy re-encryption; privacy protection.


Biographical notes: Xian Guo is currently an Associate Professor of Computer and Communication School at the Lanzhou University of Technology. He completed his BS in the Department of Mathematics at Northwest Normal University, China, 1997. He received his MS and PhD from Lanzhou University of Technology, China in 2008 and 2011, respectively. His research interests include future internet architecture and wireless networking security.

Cheng Chen is currently a Master Student in Computer and Communication School at the Lanzhou University of Technology. He completed his Bachelor degree in Computer Science from the Lanzhou University of Technology in 2013. From 2014, he started his Master in Computer Science. His research interests are future internet architecture and security of the wireless network.

Ma-Jiang Zhang is currently a Master Student in Computer and Communication School at the Lanzhou University of Technology. He got his Bachelor degree in 2014 at Anhui University of Technology and started his Master studies in 2015. His research interests are future internet architecture and security of the wireless network.

Aristide Ngaboyindekwe is currently a Master student in Computer and Communication School at the Lanzhou University of Technology. He received his Bachelor degree in Electronics and Communication Systems Engineering from the National University of Rwanda in 2012. From 2016, he started his master studies. His research interests are future internet architecture and privacy and security in wireless networks.

Copyright © 2018 Inderscience Enterprises Ltd.
1 Introduction

Today’s internet architecture, Transfer Control Protocol/Internet protocol (TCP/IP) is host-centric networking, which cannot meet the requirements of some emerging Internet applications, such as content distribution, mobile application, security, etc. CCN/NDN (content-centric networking/named data networking) is a new and most competitive future Internet architecture (PARC CCN Project, UCLA NDN Project). The goal of this novel architecture is to provide a network infrastructure service that can be better suited to today’s network applications. NDN projects are paying wide attention to both academia and industry at home and abroad (Xylomenos et al., 2014; Moiseenko and Oran, 2016; Muhammad et al., 2016).

CCN/NDN communication model (PARC CCN Project, UCLA NDN Project) uses a mechanism naming ‘content/data’ instead of naming ‘host’ in TCP/IP communication model, which can effectively solve various issues that incurred by IP addressing method based on location. The naming system plays the most important role in the CCN/NDN, like IP addressing in traditional Internet architecture. A content provider can publish content that produced to the network at certain naming rules such as a flat naming mechanism or hierarchical naming mechanism. Thus, any content consumer can use a content name to request interested content from CCN/NDN network. Besides, ‘in-network caching’ mechanism is another important feature in CCN/NDN communication model. That is to say, the new communication model requires all of the routers in the network to save forwarded content. The advantage of in-network-caching is that a content consumer can directly get interesting content from closer routers. So, this scheme can effectively improve the efficiency of content distribution.

However, the scheme that content can be cached in any router forwarded the content can incur separation of content and a content producer. So, traditional end-to-end authentication mechanism used in TCP/IP maybe is unavailable. While security measures are introduced in CCN/NDN communication model, for example, the signature is required when a content provider publishes new content, security issue in CCN/NDN is still a significant problem (Tourani et al., 2017; AbdAllah et al., 2015; Chaabane et al., 2013). Recently, they make a comprehensive analysis of privacy issues in CCN/NDN. In addition, in-network caching is a key feature of reducing network flow and improving the transmission speed for CCN/NDN, but when a cache of a network node is frequently requested, an attacker can infer who the content consumer is by using timing information. Also because the publicly available content is not encrypted in CCN/NDN, some intermediate nodes on the network can easily check the content in its cache. Furthermore, to subscribe the interested content, a content consumer must use a content name that is always semantically related to the subscribed content, so, an attacker can easily infer what content a consumer is interested in by monitoring the content request information. Finally, because the digital signature in data packet must be publicly verifiable, an attacker also can easily capture the identity of the content signer according to the signature information. These privacy issues mentioned above are known as cache privacy, content privacy, name privacy and signature privacy.

Aiming at security problems in CCN/NDN, secure content delivery mechanisms are proposed in Wood and Uzun (2014), Misra et al. (2013) and Guo et al. (2016). These schemes can be applied in different environments, and they can ensure that only a legitimate network user can get content from the network or publish content to the network. However, the privacy issue is not considered in these papers. A privacy protection mechanism based on attribute encryption algorithm is proposed in Da Silva and Zorzo (2015), which a proxy server needs to be added to the routing node of CCN/NDN network. The proxy server can withdraw a content consumer’s rights at any time to protect the privacy information of a content provider and any content consumer. However, the proposed scheme in this paper cannot prevent some attacks such as denial of services (DoS) attack, route hijacking attack and it increases the computational overhead of a routing node. Based on the theory of information entropy, a solution that resolves cache privacy problem is proposed in Ge et al. (2015). However, the proposed scheme in the paper increases the time that a content consumer request content. So, it can make the advantage of low latency for cache strategy ineffective and also can make the additional computation overhead higher. In Nabeel et al. (2012), based on Paillier homomorphic cryptosystem (PHC; Paillier, 1999), a privacy protection scheme for the publish-subscribe system is proposed. However, the scheme is based on the IP network architecture, which is not suitable for NDN.

In this paper, to solve problems of name privacy and secure content transmission, based on homomorphic cryptography (Paillier, 1999) and proxy re-encryption (PRE; Blaze et al., 1998), we propose a privacy-aware transmission scheme for NDN (PATNS_NDN). In our new scheme, a quadratic function is first constructed to generate an ‘alias’ of a content name. Thus, any content consumer
can subscribe the interested content with the content alias. But a content provider still uses the content name to publish the generated content. Furthermore, because the blinding algorithm that proposed in Nabeel et al. (2012) is used to blind the content alias in interest packet and the content name in the data packet, our scheme can efficiently solve the problem of name privacy in NDN. In addition, the only legitimate content consumer can access content on the network because homomorphic PRE mechanism is used to achieve secure content transmission. It has been proved that our solution can satisfy the following security attributes such as data confidentiality, data authentication and name privacy.

The rest of this paper is organised as follows. In Section 2, we review the related works. In Section 3, we introduce preliminaries. In Section 4, we introduce the system model used in this paper. In Section 5, we describe the details of our proposed scheme. In Section 6, we informally analyse security attributes of our scheme. In Section 7, the efficiency analysis is discussed. In Section 8, we draw our conclusions.

2 Related works

2.1 PRE-SCD protocol

In Wood and Uzun (2014), based on PRE (Blaze et al., 1998) and identity-based cryptography, a secure content transmission scheme PRE-based secure content distribution (PRE-SCD) is designed. In scheme PRE-SCD, a content source encrypts the produced content using the identity. All content on the network would be encrypted with the PRE cryptography so that the scheme has the following advantages:

- in-network caching can be effectively used
- the problems such as key and content leakages can be prevented
- the solution of key management can be simplified
- can improve overall security through end-to-end encryption between a content consumer and a content producer.

With an architecture-based entirely on PRE for content protection, each piece of content would be encrypted once by the producer using the identity of the respective content name. The content producer is the only entity that can provide and save the relative secret key for this identity, so, when the content is distributed throughout the network, it remains secure. Note that when any user is interested in some content, it must request a corresponding re-encryption key from the content source after it receives the interested and encrypted content. Until it re-encrypts, the encrypted content that comes from the original content provider or the network using the secret key of the respective itself identity, the content consumer will decrypt and capture the expected content. It is clear that any network user can publish the produced content to the network.

However, in the scheme Wood and Uzun (2014), to ensure that only a legal content consumer can get the re-encryption key and further capture the content from the network, an end-end secure authentication process is needed between a content producer and a content consumer even if the content consumer has downloaded the encrypted content from a neighbouring router that stored the content. So, it is the only method to satisfy this requirement that the content source is always required online. But this requirement could be difficult in some environments that do not rely on trusted CDNs or traditional PKI cryptosystems.

2.2 BE-SCD protocol

Broadcast encryption (Fiat and Naor, 1994) is a cryptographic technique that extensively used in secure content transmission. For example, a content provider can encrypt and publish some content to a user subset on a specific channel. Any user in the subset can then use their secret share to decrypt the encrypted content. Based on broadcast encryption, a secure content distribution for ICN (BE-SCD) was given. In BE-SCD, a content provider (e.g., Netflix, etc.) first uses a secret key to encrypt the publishing content and then produces an Enabling Block. The Enabling Block carries the materials will be used by an authorised user to extract the secret key. Finally, the content provider publishes the encrypted content and Enabling Block to the network. In Misra et al. (2013), the authors assume a hierarchical set-up. So, the encrypted content flows from the content producer to the Content Delivery Network nodes, to the internet service provider (ISP), and finally to the end user who requests the content. Apparently, PRE-SCD needs to rely on the internet infrastructure or Content Delivery Network.

In Misra et al. (2013), when some content consumer receives a content packet from its adjacent router, it firstly computes the decryption key that decrypts the content packet using its' secret share and the information contained in Enabling Block. The consumer can capture the interested content once it gets the decrypted key. It has been proved that only a legitimate user can access content on the network.

In PRE-SCD, the solution, which the encrypted content is sent along with the materials used for calculating the decrypted key by the content consumer, can efficiently resolve the problem that the content publisher is always required online in PRE-SCD. However, this scheme requires that the content provider pre-distributes the secret share to each user in the registration stage. Therefore, it is not fit for the multi-source environment that allows any source has right to publish content to the network. And this paper mainly focuses on such network that neither relies on CDN and/or on the internet infrastructure, such as wireless sensor network, mobile ad hoc network, etc.

2.3 ECY-CPS protocol

In Ravi (1996), a new public-key cryptographic solution Yaksha based on Rivest-Shamir-Adleman (RSA) is
Yaksha system split origin RSA private key into two parts. One portion is considered as a user’s Yaksha private key, and the other one server as Yaksha server’s private key. Yaksha system often is used to achieve cryptographic functions such as joint digital signature, key escrow, key exchange, etc.

In Guo et al. (2016), based on elliptic curve cryptography (ECC; Koblitz, 1987), that maybe provide same security strength for a far shorter key and reduce processing overhead compared with RSA, firstly proposed a new Yaksha system and named it as EC-Yaksha system. And then, ECY-CPS (content publish/subscribe system on EC-Yaksha) is developed. In this new scheme, EC-Yaksha server manages joining and leaving of a node in the network and is responsible for monitoring content publishing and content subscribing. Namely, this scheme only allows the legitimate user who has ‘licence’ to publish or subscribe content. Here, the licence is, in fact, a joint signature that the content consumer and the EC-Yaksha server jointly generate. The primary goals of ECY-CPS are as follows;

- it can be fit for the multi-source environment
- the EC-Yaksha server that is independent of the producer is used, so, the requirement that the content source must always be online is unnecessary
- ECY-CPS may not rely on the internet infrastructure or CDN.

Finally, using the joint signature results in preventing the content pollution attack and the interest flooding attack in ECY-CPS.

2.4 ABE-PPS protocol

The concept of attribute encryption is first proposed in Sahai and Waters (2005). The attribute encryption is a one-to-many encryption scheme. To perform fine-grained access control on the encrypted information, a threshold access structure is used. Based on attribute encryption, a privacy protection system is designed in Da Silva and Zorzo (2015), this system is called as ABE-PPS. In ABE_PPS, a subscriber can decrypt the message encrypted by a publisher if and only if a number of same attributes that possessed by the subscriber and the publisher are greater than or equal to the system threshold that this system specified.

In ABE-PPS, NDN routers need to play a role of the proxy server. A user firstly needs to register to the system and obtain a set of attributes. When a user subscribes content from the network, it must send two interest packets. One of the two interest packets is used to subscribe the interested content, and another one is used to forward the access structure. A proxy can only use the second interest packet. The proxy can check the access structure of the user, but cannot decrypt it. In ABE-PPS, a proxy can revoke the access privilege of one or more users in real time to protect the identity information of the content publisher and the privacy of the content. The proxy will renew the key of a proxy server for every revocation. However, this scheme will cause DoS attack or hijack attack, and these attacks can affect the whole area of the attacked routers, and has a significant impact on the network and increase the computation cost of the routers.

2.5 CC-PPS protocol

In Ge et al. (2015), a collaboratively caching strategy of privacy protection for CCN was proposed. We name this scheme as CC-PPS. In CC-PPS, the concept of information entropy is introduced. Information entropy is used to express the occurrence probability of some particular information. To measure privacy protection from the perspective of information entropy, this scheme can increase the uncertainty of the cache content storage by building an anonymous space region and expanding the search scope. The larger probability of requesting content is, the smaller the uncertainty reduction amount is, the less the privacy information can be leaked. Therefore, it can reduce the privacy information disclosure by confusing storage according to the popularity of the content, and it can enhance the privacy protection of consumers by increasing the uncertainty that an attacker performs an attack. At the same time, the scheme can be effective against cache pollution and flood attacks.

In this scheme, each router constructs an anonymous region, which increases the router’s hardware requirements. Due to confusing storage in CCN, the time for requesting content by consumers becomes longer, which will invalidate the advantage of low latency brought by network caching strategy in CCN, and increase the additional computing overhead.

2.6 PHC-CPP protocol

Homomorphic cryptosystem has been extensively used in many security solutions (Ozdemir and Xiao, 2011; Han and Xiao, 2016; Ozdemir et al., 2015) PHC (Paillier, 1999) is a kind of additively homomorphic asymmetrical cryptosystem with semantic security. PHC is mainly used in big data, cloud computing and so on. A privacy protection scheme based on PHC is proposed in Nabeel et al. (2012) to publish content to Brokers or subscribe content from Brokers. We call this scheme as PHC-CPP in short. In this scheme, a content subscriber negotiates a shared key with a content publisher. The publisher encrypts the produced content by using PHC and blinds the content name. A subscriber subscribes the interested content from the Broker by using the blinded information of content publisher. When a broker receives the subscription information, it performs the matching decision, and then sends the successful matching encrypted content to the subscriber. This scheme not only ensures the privacy of publishers and consumers but also prevents Brokers from learning the publisher’s rules of encryption and blindness.

Brokers are fixed in this scheme, and a subscriber subscribes the interested content from the specified Broker. As the content amount stored in Brokers increase,
the difficulty and computational overhead of Brokers’ matching operations will also increase, at the same time, the latency of obtaining a requested content will become longer. Above all, this publish-subscribe system is based on IP network architecture. It is not suitable for NDN architecture.

3 Preliminaries

3.1 CCN/NDN overview

In new architecture CCN/NDN (Figure 1), content retrieval is driven by the content consumer. That is to say; the content consumer uses an interest packet to request the interested content by using the content name. Intermediate nodes that have received an interest packet will firstly choose one or more interfaces from their outgoing faces, and then forward the interest packet to next hop node. Eventually, the provider that maybe is the content producer or some network node who forwarded the requested content will reply with a named data packet. Note that the data packet is self-certification. That is to say; the data packet will carry data integrity and authentication information according to receiving an interest packet. So, this feature enables in-network caching scheme. In CCN/NDN, the network node will route the data packet back to the consumer according to the ‘bread crumbs’ routing that left by the interest at intermediate nodes. To implement the above communication process, CCN/NDN node requires three basic components. They are, respectively, the pending interest table (PIT), the forwarding information base (FIB) and the content store (CS). Here, CCN/NDN uses the PIT table to record the pending interest packets. So, a corresponding ‘pending interest’ entry will be added to the PIT table, when an intermediate node has forwarded an interest packet through its some outgoing face. The FIB table is used to store name prefixes (domains) and their corresponding outgoing face. In CCN/NDN, CS is a content repository that used to store the forwarded content by the network node. You can refer to PARC CCN Project, UCLA NDN Project in detail.

3.2 PRE overview

Proxy re-encryption (PRE) (Blaze et al., 1998) is a significant public key cryptographic scheme. The main idea of this scheme is that it allows an untrusted proxy to transform a ciphertext encrypted under Alice’s public key to the one encrypted with Bob’s public key, given a re-encryption key provided by Alice. The detailed description about PRE is as follows.

Formally, the PRE scheme mainly has six components: (Setup, KeyGen, Encrypt, ReKeyGen, ReEncrypt, Decrypt) defined below in general terms for multi-hop schemes. A single-hop scheme only permits a piece of ciphertext to be re-encrypted once. In this context, a level 1 ciphertext is the original encrypted ciphertext; a level 2 ciphertext is the result of a level 1 ciphertext being again re-encrypted, and so on.

Figure 1 NDN communication model (see online version for colours)

Setup(\(t^k\)): This algorithm needs a security parameter \(k\). The value \(k\) determines the size of the underlying group upon which all operations take place, and then generates and outputs the public set of parameters \(params\). This procedure may also output a master secret key.

KeyGen(params, \(pk_i, \ m\)): Encrypts plaintext \(m (m \in M)\) using the input public key \(pk_i\) (or identifier) and outputs the resulting level one ciphertext \(c_i^1\).

ReKeyGen(params, \(pk_i, \ pk_j\)): Generates and outputs a re-encryption key \(rk_{ij}\) using the public parameters and public keys for users \(i\) and \(j\).

ReEncrypt(params, \(rk_{ij}, \ c_i^j\)): Re-encrypt the level \(n\) ciphertext \(c_i^n\), which is encrypted with the public key of user \(i\), to a new level \(n + 1\) ciphertext \(c_i^{n+1}\) using the re-encryption key \(rk_{ij}\) that may then be decrypted by the secret key of user \(j\).

Decrypt(params, \(sk_j, \ c_i^{n+1}\)): Parse the level \(n\) ciphertext \(c_i^j\) to determine \(n\), decrypt the ciphertext accordingly using the secret key \(sk_j\), and output the original plaintext \(m\).

PRE can be classified into two main types: unidirectional or bidirectional. In a unidirectional scheme, a ciphertext originally produced by Alice and then transformed by the untrusted proxy for Bob cannot be transformed back to the one for Alice.
3.3 Variant of PHC

PHC (Paillier, 1999) is a public-key cryptosystem based on the composite residuosity class problem. PHC also is a homomorphic cryptosystem. So, it satisfies two attributes such as additive homomorphic and multiplicative homomorphic. In Nabeel et al. (2012), PHC is redesigned and blinding algorithm based on PHC is proposed. Below we describe the detail of PHC and its variant according to Paillier (1999) and Nabeel et al. (2012).

1 Key generation algorithm Gen()

In this algorithm of key generation of PHC, randomly select two large prime numbers \( p \) and \( q \). Let \( n = pq \) and \( \lambda = \text{lcm}(p-1, q-1) \). That is to say, \( \lambda \) is the least common multiple of \( p-1 \) and \( q-1 \). Randomly select a base \( g \in Z(n^2) \) such that the order of \( g \) is a multiple of \( n \). So, a \( g_p \) can be efficiently found by randomly choosing \( g_p \in Z(n^2) \), then verifying the following condition for \( u \in \mathbb{Z}_n = \{u < n^2 | u = 1(\text{mod} n)\} \).

\[
gcd(L(g_p^2(\text{mod} n^2), n)) = 1, \text{ where } L(u) = (u-1)/n. \tag{1}
\]

In this case, set \( \mu = L(g_p^2(\text{mod} n^2))^{-1}\text{mod} n \). The public encryption key is \((n, g_p)\). The secret decryption key is \((\lambda, \mu)\), or equivalently \((p, q, \mu)\).

2 Encryption algorithm Enc(, )

For plaintext message \( m \in \{0, 1, ..., n-1\} \), selects a random number \( r \) in set \( \{1, 2, ..., n-1\} \), and uses a public key \((n, g_p)\) and encrypts \( m \) as \(\text{Enc}(m, (n, g_p), r) = g_p^m \cdot r^\mu(\text{mod} n^2)\).

3 Decryption algorithm Dec( )

Given a ciphertext \( C, \) here \( c \in \mathbb{Z}/(n^2)^\#, \) decryption algorithm \(\text{Dec}(\lambda, \mu, C)\) outputs the original plaintext \( m \).

\[
m = \text{Dec}(\lambda, \mu, C) = (L(C^\lambda \text{mod} n^2) \cdot \mu) \text{mod} n. \tag{2}
\]

4 Digital signature algorithm Sig(, ) and Ver (, )

We assume that the public key of any user \( X \) is \((n, g)\) and the secret key is \((\lambda, \mu)\). X signs on \( m \) with a secret key and gets signature \( (S_1, S_2) = \text{Sig}_X((n, g), (\lambda, \mu), m) \), where \( S_1, S_2 \) are as follows:

\[
S_1 = L(h(m)^\lambda \text{mod} n^2) \text{mod} n
\]

\[
S_2 = (h(m) \cdot g^{-S_1})^{(1/n)\text{mod}\lambda} \text{mod} n. \tag{4}
\]

Signature verification algorithm \( \text{Ver}(, ) \) is shown as follows:

\[
\text{Ver}(g \cdot \text{Sig}_X(m)) = \text{True}
\]

\[
h(m) = g^S \cdot S_2 \text{mod} n^2. \tag{5}
\]

5 Blinding algorithm \( \text{Blind}(, ) \)

In this paper, the blinding algorithm \( \text{Blind}(, ) \) based on PHC proposed in Nabeel et al. (2012) is simplified. The simplified blinding algorithm is as follows.

In this algorithm, firstly selects secret values pair \((e, d)\), where \( e + d = 1(\text{mod} n^2) \) for blinding. And then, we assume that the public key of some user is \((n, g)\) and its secret key is \((\lambda, \mu)\). In addition, selects a random number \( r \in Z^\#_n \), and define a value \( \varphi(\varphi \in Z_n) \) that used to match verification between any two blinding values.

Below, we illustrate the blinding algorithm of Nabeel et al. (2012) by an example. For two values \( m_a \) and \( m_b \), the process that it blinds the two values is as follows:

\[
\text{Enc}(m_a, (n, g_p), r) = g^{m_a} \cdot r^\mu(\text{mod} n^2) \tag{6}
\]

\[
\text{Blind}(\text{Enc}(m_a, (n, g_p), r)) = g^{e \cdot (\text{Enc}(m_a, (n, g_p), r))} \cdot \text{Enc}(\varphi(n, g_p), r)^d \text{mod} n^2 \tag{7}
\]

\[
\text{Enc}(-m_a, (n, g_p), r) = g^{m_a} \cdot r^\mu(\text{mod} n^2) \tag{8}
\]

\[
\text{Blind}(\text{Enc}(-m_a, (n, g_p), r)) = g^{e \cdot (\text{Enc}(-m_a, (n, g_p), (r)))} \text{mod} n^2. \tag{9}
\]

The result of binding \( m_a \) and \( m_b \) is respectively \(\text{Blind}(\text{Enc}(m_a, (n, g_p), r)) \) and \(\text{Blind}(\text{Enc}(-m_a, (n, g_p), r))\).

6 Match algorithm \( \text{Match}(, ) \)

In Nabeel et al. (2012), a match algorithm also is defined to make a match operation between the two blinded values. We assume that \( b_1 \) and \( b_2 \) are two blinded values, where \( b_1 = \text{Blind}(\text{Enc}(m_a, (n, g_p), r)) \)

\[
b_2 = \text{Blind}(\text{Enc}(\varphi - m_a, (n, g_p), r)) \tag{10}
\]

Then, the match function \( \text{Match}(, ) \) is described as follows:

\[
\text{Match}(b_1, b_2) = L((b_1, b_2) \text{mod} n^2) \cdot \mu \text{mod} n. \tag{11}
\]

Make,

\[
y = (b_1, b_2) \text{mod} n^2
\]

\[
y = \text{Blind}(\text{Enc}(m_a, (n, g_p), r)) \cdot \text{Blind}(\text{Enc}(\varphi - m_a, (n, g_p), r)) \text{mod} n^2 \tag{11}
\]

\[
y = (g^{e \cdot (\text{Enc}(m_a) \cdot \text{Enc}(\varphi - m_a)))^d(n, g_p), r) \text{mod} n^2
\]

\[
y = (\text{Enc}(\varphi)^d \text{mod} n^2. \tag{12}
\]
Then,
\[ \text{Match} = L(y) \cdot \mu \mod n = \varphi. \] (13)
It indicates that the match succeeds if the matching value of function \( \text{Match}() \) equal to \( \varphi \), otherwise, the matching fails.

### 3.4 Quadratic function

To generate an alias of a content name, a quadratic function is defined in this paper. The basic expression of this quadratic function is \( f(x) = ax^2 + bx + c \) \((a \neq 0)\). The image of the quadratic function \( f \) is an opening up or down parabola. The symmetry axis of this parabola is a line \( x = -\frac{b}{2a} \). The vertex of the quadratic function is \( \left(-\frac{b}{2a}, \frac{4ac-b^2}{4a}\right) \). In this paper, we assume that the identity information value of any user can be decomposed into product of two numbers, and the two obtained numbers are, respectively, the value of coefficients \( a \) and \( b \) for the constructed quadratic function. And then select a random number as the value of coefficient \( c \). For example, we assume that the identity information value of a user is 336. It is clear that this 336 can be decomposed into two values 16 and 21. So, the value of coefficient \( a \) for the new quadratic function is 16, the value of coefficient \( b \) is 21. And then we randomly select a number 15 as the value of the coefficient \( c \). \( f(x) = 16x^2 + 21x + 15 \) is our constructed quadratic function.

### 4 System model

#### 4.1 Network model

To illustrate the PATS_NDN scheme, a simplified NDN network model is shown in Figure 2. Assuming the network node \( CP \) (data source) is a content provider. That is to say; \( CP \) is a data source. Node \( CC_i \) \((i = 1, 2, \ldots, n)\) are users. That is to say, \( CC_i \) is the content consumer. Any user must register with a network manager (such as ISP) before joining the network. The network manager distributes required cryptographic materials to legitimate users. When any user leaves the network, the network manager will revoke the relevant information. In this paper, we assume that the content provider plays a role of the network manager. \( N \) is a network consisting of NDN routers. Of course, in a wireless environment, the router in core network \( N \) could be in a general node. They play two kinds of roles such as router and host. Simplified NDN network model is based on the CCN/NDN communication model, which is required to forward the interest packet and data packet according to the rules of CCN/NDN, and routers in the network are capable of caching the forwarded data packet.

#### 4.2 Threaten model

In our threaten model, we assume that routers in network \( N \) are honest but curious. Namely, all of them will operate honestly according to the rule of the NDN network model. They will save a received data packet in the content storage, forward interest/data packet and do the matching operation. They may also be curious to obtain the metadata in a forwarding interest/data packet or data packet stored in the content storage. In short, routers in NDN are not credible regarding confidentiality, but rather regarding storing and managing the content message. In this paper, we mainly focus on the outside attacker. We assume that this type attacker only can eavesdrop the interest/data packet transmitted on the network. Moreover, they may be also initiate some active attacks such as tampering some packets or replaying some out of date packets. In addition, they can also learn what a consumer is interested in or what the content provider produces by analysing the interest and data packets.

**Figure 2** Simplified NDN network model (see online version for colours)

#### 4.3 Security requirements

In our system, to block malicious acts from the attackers above and ensure the privacy and security of the interest/data packet delivery, we require that our new solution can satisfy the security attributes described below.

*Data confidentiality:* The attribute requires that only legitimate network user has right to access the interest/data packet on the NDN network, and the illegal one should be prevented from capturing the relevant encryption key on the packet. Here, the encryption key includes the shared key between the consumer and the content provider and PRE key and so on.

*Data authentication:* The attribute describes the message integrity and authentication about the network user identity. The former guarantees any altered data during transmission could be detected. These data are often the content name in the interest packet and the content of the content packet. The latter means that only the legitimate network user can broadcast the interest packet to the network for subscribing the interested content from the network and publish the content to the network.

*User privacy:* An attacker can intercept and check the metadata in interest and data packets. We want to operate on the message in the interest packet and data packet so that the attacker cannot get any user’s private information. At the same time, although a routeing node can correctly execute
the matching operation between a received interest packet and a stored data packet, it cannot learn any information about user’s privacy.

5 PATS_NDN systems

Participating entities in our PATS_NDN scheme include content consumers $CC_i$, one content provider $CP$, and core network $N$ consisted of routers (in a wireless environment, they are a general node) in NDN. We assume that any consumer $CC_j$ acquires the master secret key $K_{CC-CP}$ shared with the $CP$ when it registers to the content provider $CP$. Content provider publishes generated content to network $N$ by using data packet. Content consumer, who is interested in some content, firstly encrypts this content name with a shared key, and then sends to the content provider for generating an alias of the content name and binding the generated alias. When the content provider receives a message that requests an alias from a content consumer, it firstly constructs a quadratic function used to calculate an alias of the content name that the consumer is interested in according to identity information and generates a re-encryption key according to the public key of the consumer. And then, the content provider uses the blind algorithm (in Section 3.3) to blind the alias generated and encrypts the blinded alias and the re-encryption key with a shared key between $CP$ and $CC_i$. Lastly, $CP$ sends generated ciphertext to the content consumer. A content consumer generates an interest packet based on the information received from the content provider, and then sends this interest packet to the network $N$ for subscribing interested content. Finally, when some router in network $N$ receives an interest packet, it verifies if there exists content matching with the interest packet in its content storage. If verification successes, then it forwards the corresponding content package to the consumer, otherwise, it continuously forwards the interest packet according to the FIB. When a content consumer receives the content packet, it will extract the content from the content packet. The detail of our solution is described as follows.

5.1 Initialisation

In the initialisation phase of the PATS_NDN scheme, all parameters $\{p, g, k, r, g\}$ used in this scheme are firstly selected, and the key generation algorithm of PHC (in Section 3.3) is used in our scheme to generate public/secret key pairs for all users in the network. For example, we assume that public key and a secret key of any user $X$, respectively, are $(n_X, g_X)$ and $(\lambda_X, \mu_X)$. In our scheme, the public key $(n_X, g_X)$ of user is managed by trusted third party (TTP), and the secret key $(\lambda_X, \mu_X)$ is saved by the user. Finally, when content consumer $CC_i$ registers to content provider $CP$, they negotiate cryptographic materials, such as shared key $K_{CC-CP}$, and the cryptographic algorithms used between $CC_i$ and $CP$.

5.2 Content publishing

5.2.1 Content name blinding

Whenever content provider $CP$ produces new content $m$ that its name is $m_{name}$, it will process as follows:

- It firstly generates a secret value $(e, d)$ and selects a random number $r_m (e, d \in Z^n)$. In this secret value pair $(e, d)$, $e$ and $d$ must satisfy the conditions $e + d = 1 (mod n^2)$, where $e$ is used for blinding content name, $d$ is used for blinding content alias in the future.
- And then, a content provider $CP$ generates a matching success value $\varphi (\in Z^n)$. This value will be used when any router in network $N$ makes a matching operation between a received interest packet and data packets stored in the content storage.
- Finally, the content provider $CP$ constructs a quadratic function $f_{cp}(.)$ by using own identity information and random number $r_m$ generated in equation (1). It then calculates a function value $f_{cp}(m_{name})$ of the content name $m_{name}$, and uses the blinding algorithm for the variant of PHC to blind the content name $m_{name}$. The blinding process of $CP$ is as follows.

$$Enc(f_{cp}(m_{name})) = g_{cp}^{e_{m_{name}}} \cdot r_m^{\varphi} \mod n_{cp}^2$$  \hspace{1cm} (14)

$$Blind_{cp}(Enc(f_{cp}(m_{name}))) = g_{cp}^{e_{m_{name}}} \cdot Enc(\varphi)^{\varphi} \mod n_{cp}^2 \hspace{1cm} (15)$$

The value $Blind_{cp}(Enc(f_{cp}(m_{name})))$ generated in equation (15) is the blinding value of the content name $m_{name}$. The content provider stores this information such as content $m$, $m_{name}$, $f_{cp}(m_{name})$, and the secret value $(e, d)$.

5.2.2 Content encryption and signature

- Content encryption

The content provider $CP$ encrypts a produced content message $m$ with its own public key $(n_{cp}, g_{cp})$ by using encryption algorithm of PHC. The obtained ciphertext $C_{cp}$ is as follows:

$$C_{cp} = Enc((n_{cp}, g_{cp}), m) = g_{cp}^{n_{cp}} \cdot r_{cp}^{\nu_{cp}} \mod n_{cp}^2$$  \hspace{1cm} (16)

- Content signature

The content provider $CP$ signs on a produced content message $m$ with its own secret key $(\lambda_{cp}, \mu_{cp})$ by using the signature algorithm of PHC. This operation generates
the signature value pair \( \text{Sig}_{c_{cp}} = (S_{c_{cp}}, S'_{c_{cp}}) \), where \( S_{c_{cp}} \) and \( S'_{c_{cp}} \) can be obtained as follow:

\[
S_{c_{cp}} = \frac{L((\text{Hash}(m))^k \mod n_{cp})}{L(g_{cp}^k \mod n_{cp})} \mod n_{cp} 
\]

\[
S'_{c_{cp}} = (\text{Hash}(m)) \cdot g_{cp}^{S_{c_{cp}}} \mod n_{cp}. \tag{18}
\]

Now, the content provider \( CP \) can produce a content packet \( \{\text{Blind} \_c \_cp \_m \_c \_cc \_name \_alias, \text{Enc} \_m \_c \_cc \_name \_alias \} \) and publishes the data packet to the network \( N \).

### 5.3 Content subscribing

#### 5.3.1 Content alias request

Assume that any content consumer \( CC \) is interested in content that name is \( m_{\text{name}} \). To acquire the alias of \( m_{\text{name}} \) from the content provider \( CP \), the content consumer \( CC \) generates and sends the following information to the content provider \( CP \):

\[
C_{\text{name}} = E_{k_{cc-cp}}(m_{\text{name}} \| ID_c). \tag{19}
\]

where \( K_{cc-cp} \) has shared a key between \( CC \) and \( CP \), \( ID_c \) is the identity of the consumer \( CC \), \( E(.) \) is symmetrical encryption algorithm negotiated between \( CC \) and \( CP \) in initialisation stage.

#### 5.3.2 Generate and blind content alias

When content provider \( CP \) receives a message \( C_{\text{name}} \) from the content consumer \( CC \), it attempts to decrypt the message \( C_{\text{name}} \) with shared key \( K_{cc-cp} \).

- If content provider \( CP \) cannot decrypt the \( C_{\text{name}} \), then it thinks that the message comes from an illegal user and discards \( C_{\text{name}} \).
- Otherwise, according to the identity information \( ID_c \) of the content consumer \( CC \), point \( (m_{\text{name}}, f_{cp}(m_{\text{name}})) \) stored by the content provider \( CP \), the content provider \( CP \) constructs a quadratic function \( f_{cc}(.) \) and calculates a point \( (m_{\text{alias}}, f_{cc}(m_{\text{alias}})) \) in \( f_{cc} \). Here, the point \( (m_{\text{alias}}, f_{cc}(m_{\text{alias}})) \) must satisfy the two conditions \( m_{\text{alias}} \neq m_{\text{name}}, f_{cc}(m_{\text{alias}}) \neq 0 \). Note that \( m_{\text{alias}} \) is the alias of content \( m \) that name is \( m_{\text{name}} \).
- Now, content provider \( CP \) generates a re-encryption key \( r_{\text{key}} \) for content consumer \( CC \) by using the re-encryption key generation algorithm \( \text{ReKeyGen}() \) introduced in Section 3.2.
- And then, the content provider \( CP \) uses the blind algorithm to blind the alias \( m_{\text{alias}} \) of content \( m \). The blind operation is as follows:

\[
\text{Enc}(-f_{cc}(m_{\text{alias}})) = g_{cp}^{f_{cc}(m_{\text{alias}})} \cdot r_{\text{key}}^d \mod n_{cp}^2 \tag{20}
\]

\[
\text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}}))) = g_{cp}^{d} \cdot \text{Enc}(-f_{cc}(m_{\text{alias}}))^{-b_{cp}} \mod n_{cp}^2. \tag{21}
\]

where \( d \) in equation (21) is \( d \) of secret value \((e, d)\) stored by \( CP \). \( \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}}))) \) is the blinding value of alias \( m_{\text{alias}} \) of the content \( m \).

- Next, content provider \( CP \) uses the signature algorithm of PHC introduced in Section 3.3 to sign on \( \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}}))) \) with its own secret key \((\lambda_{c_{cp}}, \mu_{c_{cp}})\) and gets the following signature:

\[
(S_{c_{cp}}, S'_{c_{cp}}) = \text{Sig}_{c_{cp}}((\lambda_{c_{cp}}, \mu_{c_{cp}}), \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}})))) \tag{22}
\]

where \( S_{c_{cp}} \) and \( S'_{c_{cp}} \) are as follows:

\[
S_{c_{cp}} = \frac{L((\text{Blind} \_c \_cc \_name \_alias(\text{Enc}(-f_{cc}(m_{\text{alias}}))))^k \mod n_{cp}}{L(g_{cp}^k \mod n_{cp})} \mod n_{cp} \tag{23}
\]

\[
S'_{c_{cp}} = (\text{Enc}(-f_{cc}(m_{\text{alias}}))) \cdot g_{cp}^{S_{c_{cp}}} \mod n_{cp}. \tag{24}
\]

- Finally, the content provider \( CP \) uses the shared key \( K_{cc-cp} \) to encrypt the re-encryption key \( r_{\text{key}} \), alias \( \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}}))) \), and the signature \((S_{c_{cp}}, S'_{c_{cp}})\) and generate the following ciphertext:

\[
C'_{\text{name}} = E_{k_{cc-cp}}(r_{\text{key}} \| \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}})))) \| \text{Sig}_{c_{cp}} \| ID_c \tag{25}
\]

The content provider \( CP \) sends the ciphertext \( C'_{\text{name}} \) to the content consumer \( CC \).

### 5.3.3 Subscribe content

When content consumer \( CC \) receives messages \( C_{\text{name}} \) from the content provider \( CP \), it will perform the following steps:

- It firstly decrypts the message \( C_{\text{name}} \) with the shared key \( K_{cc-cp} \) to get the alias of the content that it is interested in. If this decryption operation fails, which indicates the message is not a message that comes from the content provider \( CP \). This message is discarded.
- Otherwise, the consumer \( CC \) uses the message \((r_{\text{key}} \| \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}})))) \| \text{Sig}_{c_{cp}} \| ID_c \) to generate the following ciphertext:

\[
(r_{\text{key}} \| \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}})))) \| \text{Sig}_{c_{cp}} \| ID_c \tag{26}
\]

- And then, using the verification algorithm \( \text{Ver}() \) of PHC introduced in Section 3.3, the content consumer \( CC \) verifies the signature of the content provider \( CP \). This verification operation is \( \text{Ver}(g_{cp} \cdot \text{Sig}_{c_{cp}} \cdot \text{Blind}_c(\text{Enc}(-f_{cc}(m_{\text{alias}})))) \). If this signature verification fails, then the content consumer \( CC \) deletes the message \( C'_{\text{name}} \) from the content provider \( CP \). Otherwise, it saves re-encryption key.
that has received and generates an interest packet included the blinding value of the alias \{Blind_{i}(Enc(-f_{m}(m_{alias})))\}. The content consumer CC broadcasts the interest packet to the network N to capture the interested content m.

5.4 Content matching

When any router in network N receives the interest packet \{Blind_{i}(Enc(-f_{m}(m_{alias})))\} from some consumer, it made a matching operation between the blinding value of alias m_{alias} Blind_{i}(Enc(-f_{m}(m_{alias}))) included in the interest packet and the blinding value of a name m_{name} for some content m Blind_{i}(Enc(f_{m}(m_{name}))) from a data packet (that previously received) and stored in the content storage. If this matching success, then the router forwards the corresponding data packet included \{Blind_{i}(Enc(f_{m}(m_{name})))\|Hash(m)|C_{cp}|\|\|\|\|\} to the content consumer CC. Otherwise, it will process the interest packet according to the forwarding rule of NDN network.

5.5 Content cover

When any content consumer CC receives the data packet included the message \{Blind_{i}(Enc(f_{m}(m_{name})))\|Hash(m)|C_{cp}|\|\|\|\|\} from the network N, firstly uses signature verification algorithm of PHC introduced in Section 3.3 to verify the signature \(\text{Sig}_{g_{cp}}\) in this message. If the verification fails, then this indicates that the data packet does not come from the content provider CP, and the content consumer CC discards the data packet; Otherwise, the content consumer CC uses re-encryption key \(r_{k_{cp\rightarrow cc}}\) to encrypt the ciphertext \(C_{cc} = Enc(r_{k_{cp\rightarrow cc}}, C_{cp})\). And then, the content consumer CC uses the decryption algorithm of PHC to decrypt \(C_{cc}\) with its own secret key \(\{\lambda_{cc}, \mu_{cc}\}\) and get content \(m = Dec(\lambda_{cc}, \mu_{cc}, C_{cc})\). Finally, it verifies the integrity of the content m according to the Hash(m) value in the data packet. If this verification success, then this content m is the content that this consumer interested in.

6 Security analysis

According to the security requirements discussed in Section 4.3, we will give a detailed analysis of the security properties of our proposed scheme in this section.

Proposition 1: (1) The content provider can prove that the request of the alias for some content must come from the content consumer that claims the request; (2) Vice versa, the content consumer can also prove that the received alias of some content must come from the content provider.

Proof: (1) In the initialisation stage of our PATS_NDN, we assume that the shared key \(K_{cc\rightarrow cp}\) between any content consumer CC and the content provider CP was dispatched. When any content consumer CC sends the request of alias for some content \(X\) to the content provider CP, the content consumer CC uses the shared key \(K_{cc\rightarrow cp}\) to encrypt this request as follows:

\[ CC \rightarrow CP : E_{K_{cc\rightarrow cp}}(X \| ID_{cc}). \] (27)

So, if the content provider CP can decrypt this received request with the corresponding shared key \(K_{cc\rightarrow cp}\), then it can prove that the request comes from the content consumer that claims the request. Of course, we assume that the content consumer is an honest participant.

The proof of (2) in the proposition is similar with that of (1).

Proposition 2: Any content consumer can prove that any received content must come from the content provider and does not tamper in transmission.

Proof: The signature algorithm of PHC and Hash() are introduced in the data packet of our scheme PATS_NDN. So, if a content consumer CC who received a data packet can successfully perform the following verification (equation (28)) with the public key \((n_{cp}, g_{cp})\) of the content provider CP, then it can prove that the received data packet must come from the content provider CP.

\[ \text{Ver}(g_{cp} \cdot \text{Sig}_{g_{cp}}(m) = \text{Hash}(m)) = \text{True} \]
\[ \text{Hash}(m) = g_{cp}^{S_{cc} \cdot S_{cc}^{n_{cp}}} \mod n_{cp}^2. \] (28)

Also, according to Hash(m), the content consumer also can decide if the content is altered during transmission in network N.

Proposition 3: In new scheme PATS_NDN, content that the content provider CP published only can be decrypted by the legal participant who subscribed the content. So, the confidentiality of content transmission can be guaranteed.

Proof: In the content publishing phase, the content provider CP uses the encryption algorithm of PHC introduced in Section 3.3 to encrypt the content m that it publishes with its public key \((n_{cp}, g_{cp})\). Here, we assume that the ciphertext generated is \(C_{CP}\).

Now, if a participant wants to decrypt the ciphertext \(C_{CP}\), it must first apply for a corresponding PRE key \(r_{k_{cp\rightarrow cc}}\) from the content provider, and then it can decrypt and get the content after it re-encrypts \(C_{CP}\) with PRE key \(r_{k_{cp\rightarrow cc}}\). However, our scheme requires that the content provider only distributes the corresponding PRE key to the legal participant that only submits an application. Here, the legal participant is a user that has registered on the server.
So, our scheme PATS_NDN can provide secure content transmission.

**Proposition 4:** In our PATS_NDN, the privacy of the content consumer and the content provider can be protected. That is to say; an attacker does not know what the consumer is interested in and what content produced by the content provider even if the attacker intercepts an interest packet or a data packet in the transmission process.

**Proof:** The scheme of alias for content and the blind algorithm for a variant of PHC (in Section 3.4) are introduced in our solution. In an interest packet, the content consumer uses the content alias from the content provider to subscribe the interested content. The content alias in an interest packet and the content name in a content packet are both blinded by the provider using the blind algorithm of a variant of PHC. So, the attacker who intercepts an interest packet or a data packet cannot get any useful privacy information about the consumer and the provider.

**Proposition 5:** In our PATS_NDN, the probability that an attacker successfully forges the blinding alias and name of some content is ignorable.

**Proof:** It is not possible for an attacker to falsify a blinding alias and a blinding name. Because the content alias generated by the content provider CP is related to the identity information of the content consumer CC which requested the content, and a secret value pair \((e, d)\) that only the content provider CP known is introduced in the blinding name and the blinding alias. If an attacker wants to forge a blinding alias needs to know the identity \(ID_{cc}\) of the content consumer, then the constructed quadratic function \(f\) according to the identity of the \(ID_{cc}\), and the secret value \((e, d)\). Even if an attacker has gotten the identity \(ID_{cc}\) of a content consumer who requested some content, then it must also perform \(C_{2} \cdot C_{c_{4}} \cdot C_{t} (t \in Z_{n})\) times calculation to successfully construct the quadratic function \(f_{cc}\) and \(C_{t} \cdot C_{t} \cdot C_{t} = t^3(t-1)^3 / 16\) times calculations need to be done by the attacker to forge a blinding alias. In case that \(t\) is huge, the success rate that the attacker successfully fakes a blinding alias of some content is closer to 0 in relatively short time.

**Proposition 6:** A router in the network can correctly execute a matching operation when it receives an interest packet requested some content.

**Proof:** The matching operation of a router in the network is an operation to make a match between the blinding alias of content in an interest packet and the blinding name of content in a data packet. Here, the blinding alias is carried in an interest packet that just received, and the blinding name can be received in a data packet and saved in the content storage of the router.

So, we assume that a router receives an interest packet contained the blinding alias \(Blind_{cc}(Enc(-f_{cc}(m_{cc}))\), and then it uses function \(Match()\) to perform match operation between \(Blind_{cc}(Enc(-f_{cc}(m_{cc}))\) and some blinding name \(Blind_{cc}(Enc(f_{cc}(m_{cc}))\). The process of the match is as follows:

\[
Match(Blind_{cc}(Enc(f_{cc}(m_{cc}))))
\]
\[
\cdot Blind_{cc}(Enc(-f_{cc}(m_{cc}))))
\]
\[
= L(Blind_{cc}(Enc(f_{cc}(m_{cc}))))
\]
\[
\cdot Blind_{cc}(Enc(-f_{cc}(m_{cc})))) \mod n^2 \cdot \mu \mod n.
\]

Make,

\[
y = Blind_{cc}(Enc(f_{cc}(m_{cc}))))
\]
\[
\cdot Blind_{cc}(Enc(-f_{cc}(m_{cc})))) \mod n^2
\]
\[
y = Blind_{cc}(Enc(f_{cc}(m_{cc}))))
\]
\[
\cdot Blind_{cc}(Enc(-f_{cc}(m_{cc})))) \mod n^2
\]
\[
= g^r \cdot Enc((r \cdot f_{cc}(m_{cc}))) \cdot \phi \cdot g^d \cdot (Enc(-f_{cc}(m_{cc}))) \mod n^2
\]
\[
= g^{r+\lambda} \cdot (Enc(r \cdot f_{cc}(m_{cc}))) \mod n^2
\]
\[
= (Enc(\phi)) \cdot (Enc(r \cdot f_{cc}(m_{cc}))) \mod n^2
\]
\[
= (Enc(\phi)) \mod n^2.
\]

Then,

\[
Match = L(y) \cdot \mu \mod n
\]
\[
= \phi.
\]

So, the router can correctly execute the matching operation.

7 Efficiency analysis

7.1 Theoretical analysis and comparison

Table 1 shows the comparison results of our scheme, Wood et al.’s and Nabeel et al.’s in the following aspects: the number of cryptographic operations (NCO) and network exchange messages (NEM). Whether it relies on the server (RS), whether the ciphertext is publicly verifiable (CPV), whether it uses a shared key (SK), whether there exists a problem that it can leak a decryption key (DKL), or whether it requires the routing node to verify signature (RNVS).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>NEM</th>
<th>RS</th>
<th>CPV</th>
<th>SK</th>
<th>DKL</th>
<th>RNVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood et al.’s</td>
<td>(t + 2)</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Nabeel et al.’s</td>
<td>4</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>PATS_NDN</td>
<td>5</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

\(t\) is logged out user threshold value.
It can be seen from Table 1 that Nabeel et al.’s scheme in (2012) and our scheme both need a shared key, but the two schemes do not rely on the server, and cannot leak the decryption key. But the Nabeel et al.’s scheme only requires the signature verification but does not require the source authentication. In addition, it uses the IP network to carry on the content distribution. So, it is not suitable for the NDN network. Wood et al.’s scheme in (2014) and our scheme are both publicly verifiable, and they cannot leak the identity information of the content provider. However, Wood et al.’s scheme needs to rely on a server and can leak the decryption key. In addition, the data source uses CDN (content distribution network) to publish content.

We make a comparison between Nabeel et al.’s scheme and our scheme regarding the success rate of matching operation. The comparison results of the matching success rate during each aggregation are shown in Table 2, which involves Nabeel et al.’s scheme and our scheme. The matching success rate of our proposed scheme is higher than Nabeel et al.’s scheme because content produced by the content provider can be stored in any routing node of network $N$ in our scheme. So, a content consumer can subscribe interested content from adjacent routers by broadcasting an interest packet. At the same time, these neighbouring routers can carry out the matching operation between the interest packet and the content packet, thus result in a higher rate of matching success. However, a content consumer must subscribe interested content from specified Brokers because content can be stored in fixed Brokers in Nabeel et al.’s scheme in (Da Silva and Zorzo, 2015). So, the matching success rate is low.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Matching success rate comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Nabeel et al.’s</td>
<td>2.00%</td>
</tr>
<tr>
<td>PATS_NDN</td>
<td>5.00%</td>
</tr>
</tbody>
</table>

7.2 Simulation analysis and comparison

In this section, we analyse time efficient that a content provider produces a content packet and a content consumer extracts content from a content packet between Nabeel et al.’s scheme in Da Silva and Zorzo (2015) and our scheme PATS_NDN.

We firstly make an analysis about the time efficient that a content provider produces a content chunk. In simulation experiment, the key length that we selected is 1024B. The comparison result is shown in Figure 3. It is clear that our scheme needs an extra time to construct a quadratic function used to generate the content alias, and used to blind this alias, except that our scheme needs to perform blinding, matching and extracting. So, Figure 3 shows that PATS_NDN scheme proposed in this paper requires more time than the scheme in Nabeel et al. (2012) to generate a content chunk.

Secondly, we make an analysis of time expense that a content consumer extracts content from a content chunk. In this simulation experiment, the length of the key that we selected also is 1024B. The comparison result is shown in Figure 4. Figure 4 shows that time taken by our PATS_NDN scheme to extract content is smaller than that of Nabeel et al.’s scheme. Our scheme only needs to two operations such as re-encrypting and decrypting. However, Nabeel et al.’s scheme require a blind algorithm to judge the requested content, recovery condition, and decryption operation.

Figure 3 Time generated content chunk (see online version for colours)

Figure 4 Time recovered content from a content chunk (see online version for colours)

8 Conclusion

In this paper, to resolve name privacy and secure content transmission issues in NDN, we design privacy-aware transmission scheme PATS_NDN. In PATS_NDN, we introduce PHC and PRE mechanism and construct a quadratic function. The quadratic function is used to generate an ‘alias’ of the content name. Thus, a content consumer can use the content alias that blinded by the blinding algorithm of a variant of PHC to subscribe the interested content. The content provider publishes the generated content with the blinded content name. So, our
scheme can efficiently solve the problem of name privacy in NDN. Also, the only legitimate content consumer can access content on the network because homomorphic PRE mechanism is used to achieve secure content transmission. In future work, we will continue to discuss the issues in NDN and further improve our solutions.

Acknowledgements

This work is supported by NSFC No. 61461027, No. 61462060 and No. 61562059; Gansu province science and technology plan project under grant No. 1610RJYA0086; Overseas exchange fund for faculty of Lanzhou University of Technology. We thank the referees for helpful comments.

References


Websites

PARC CCN Project: Content Centric Networking project, PARC, Palo Alto, California, http://www.ccnx.org/

UCLA NDN Project: NSF Named Data Networking project, University of California, Los Angeles, http://www.named-data.net/